Document Version 1.2

Multi-Signature Mechanism

To protect personal account information, EON employs the Multisignature mechanism or MSM. MSM allows you to distribute privileges to sign uncommitted transactions among multiple participants considering the quorum that covers certain percentage that should be attained in order for a transaction to be signed.

Delegation

The main idea behind MSM technology is Delegation. Delegation is a way to delegate a portion of privileges to another account(s) to sign uncommitted transactions.

All new transactions should be signed (i.e. confirmed) by the required number of participants (not just by issuer) to whom privileges have been delegated considering the percent threshold. A threshold is defined by Quorum. By default, the quorum threshold is set to 100%. You may customize the quorum in a way that the overall threshold might be lower. At the same time the quorum shouldn't exceed 100%. See Quorum Concept section.

Let's say there are three accounts A, B, and C. Account A wants to secure its deposit and decided to delegate 50% of its digital signature to account B, and another 50% to account C. At the same time account A should downgrade its own privileges down to 75%. All three of them are now have 175% of the digital signature in total. Now, when account A issues a transaction, it needs either account B or account C to sign this transaction. Though accounts B and C are now delegates of the account A, they still remain self-sufficient in signing their own transactions and do not require any kind of participation of the account A whatsoever. However, there might be scenarios when account A gives privileges to account B, and account B gives privileges to account A. In this case both accounts need each other when either of them issues a new transaction. See Scenarios section.

A proper sequence should be adhered when delegating privileges. First step is to delegate privileges and the second step is to downgrade the privilege of the account that gives privileges.

Delegated privileges may be rejected by that account you want to become your delegate. See Rejecting Privileges section.

Public and Private Accounts

Sometimes you may want to make your account public. All public accounts share their seed among other participants over the entire EON network. The main idea behind a public account is to distribute privileges among those accounts you trust the most. When you make your account public, you have no privileges whatsoever to sign up transactions even that of your

own. As soon as all required delegates confirm a new transaction, a transaction considered committed and therefore added to the block.

To make an account public, you first need to delegate privileges to sign your transactions to another account(s), then you have to downgrade your own privilege level to 0 and wait for 24 hours. After the 24 hours delay, you have to execute the `./eon publication` command. But before you do this, you must ensure that your account is not a delegate of someone else. Delegates cannot be public.

Once your account had become public, your seed becomes known and might be used by your delegates to issue transactions on behalf of your account. At the same time, anyone over the network may use your seed but as long as those accounts aren't your delegates, they do not have privileges to sign up such transactions, therefore these transactions will always remain uncommitted. Though such a transaction will never be added into the block, the public key might be considered compromised.

Quorum Concept

Quorum lets you configure your MSM mode in a more granular way. With Quorum you can set the required threshold that should be attained when issued transactions are being signed up by multiple participants. Quorum also allows you to define transaction types to which the defined percent range is applied. By default, the quorum is set to 100 percents.

There are following transaction types available:

Type	**Code**	**Description**
Registration	100	Registration of a new account
Payment	200	A coin transfer between two accounts
Deposit	300	Deposit funds to participate in the generation of blocks
Delegate	400	Sets the weight of the voice for the signature
Quorum	410	Sets the quorum for transactions
Rejection	420	Refusal to participate in transaction confirmation
Publication	430	Makes an account public
ColoredCoinRegistration	500	Colored coins creation
ColoredCoinPayment	510	Colored coins transfer
ColoredCoinSupply	520	Change in the colored coin emission

ComplexPayment	600	Complex payment (complex transaction is a transaction which includes several standard payment transactions)

The Quorum command suggests the following syntax:

`>>> ./eon quorum -all 50` - Sets the overall quorum that should be attained to 50 percents.

`>>> ./eon quorum -all 50 -100 95` - Sets the overall quorum that should be attained to 50 percents and the special quorum of 95 for transactions of the Registration type. The ratio part in this example (-100 95) is not mandatory and can be omitted. Bear in mind, that the ratio percent threshold must be different from that of the overall threshold. (e.g. `./eon quorum -all 50 -100 50` is not correct).

Rejecting Privileges

A private account which you want to become a delegate can reject delegated privileges.

Scenarios

Scenario 1: Using a Private Account with Multisignature

This scenario demonstrates how you can protect your account (i.e. private seed) that is linked to a peer.

Step 1: Creating a New Account

```
>>> ./eon seed
>>> ./eon register -p <public key>
```

Step 2: Delegating Privileges

`>>> ./eon delegate -r **EON_ID_1** -p 50 - Delegating 50% of privileges to a newly created account.`

Step 3: Downgrading Personal Privilege Level

`>>> ./eon delegate -r **MY_OWN_EON_ID** -p 50 - Downgrading your own privilege level down to 50%.`

Now that you have delegated privileges to the EON_ID_1 account, you'd need its confirmation when issuing new transactions. For example, a payment transaction:

```
>>> ./eon payment -r <EON_ID> -a 10
```

Scenario 2: Using Two Private Accounts with Multisignature

This scenario is similar to the Scenario 1: Using a Private Account with Multisignature, except that it involves two private accounts that share 50% of privileges. This scenario suggests that if either of the accounts is lost, a transaction still can be confirmed.

Step 1: Creating New Accounts

```
>>> ./eon seed
>>> ./eon register -p <public key>
```

Step 2: Delegating Privileges

```
>>> ./eon delegate -r EON_ID_1 -p 50 - Delegating 50% of privileges to a newly created account.
>>> ./eon delegate -r EON_ID_2 -p 50 - Delegating 50% of privileges to a newly created account.
```

Step 3: Downgrading Personal Privilege Level

```
>>> ./eon delegate -r MY_OWN_EON_ID -p 50 - Downgrading your own privilege level down to 50%.
```

Scenario 3: Making a Private Account Public

This scenario is quite similar to the Scenario 2: Using Two Private Accounts with Multisignature one but here we use a public account approach. When using public accounts, a seed cannot be lost because it becomes known to anyone over the entire network.

Step 1: Creating New Accounts

```
>>> ./eon seed
>>> ./eon register -p <public key>
```

Step 2: Delegating Privileges

```
>>> ./eon delegate -r EON_ID_1 -p 100 - Delegating 100% of privileges to a newly created account.
```

```
>>> ./eon delegate -r EON_ID_2 -p 100 - Delegating 100% of privileges to a newly created account.
```

Step 3: Downgrading Personal Privilege Level

```
>>> ./eon delegate -r MY_OWN_EON_ID -p 0 - Downgrading your own privilege level down to 0%.
```

Step 4: Making an Account Public

After the 24 hours delay

```
>>> ./eon publication
```

If for some reasons either of the accounts is lost, a transaction still can be confirmed by another one because both have been given 100% of privileges.

Scenario 4: Using Quorum with the Private Account

Let's say we want to make some payments and secure our deposit. Here we're setting up the quorum in a way that all payment transactions require 50% threshold.

Step 1: Creating New Accounts

```
>>> ./eon seed
>>> ./eon register -p <public key>
```

Step 2: Delegating Privileges

```
>>> ./eon delegate -r EON_ID_1 -p 100 - Delegating 100% of privileges to a newly created account.
>>> ./eon delegate -r EON_ID_2 -p 100 - Delegating 100% of privileges to a newly created account.
>>> ./eon quorum -all 100 -200 50 - Setting up a common quorum to 100% while for Payment transactions we define 50% threshold.
```

Step 3: Downgrading Personal Privilege Level

```
>>> ./eon delegate -r MY_OWN_EON_ID -p 50 - Downgrading your own privilege level down to 50%.
```

In this scenario you can make a payment solely by yourself when transferring from the common account. But to make a payment from a deposit account, you need both accounts to confirm your transaction.